Step 5 – Reading and writing to/from file, Exception Handling

In the previous step, we learned how to get input from the user via console. Now we will see that we can also use Scanner to read from file. We also need to write to file with FileWriter. And we must also learn a little exception handling. We've ignored it until now, but now it's really time!

Objectives for this step:

- I can use Scanner to read from file.
- I can use FileWriter to write to file.
- Overall, I understand how exception handling works in Java.
- I understand how I can catch and throw exceptions.
- I can use a suitable class to handle a date.

This time there are no relevant chapters in the syllabus…

Here is the questionnaire for step 5 where you can tell how it went.

In Canvas you will find some files you need for this week. Get the zip in Canvas and unpack. The intro video explains where you should put the files. As you will see below, the first tasks are a slightly different introduction than the last.

Task 1

Create a new project. Add a class that has a main method.

Task 2

Create a class «Program». It is in this class that the various tasks (4, 5, 7 and 8) are to be performed.

Task 3

In your main method; create an object of the Program class. You check if the methods you create (for problem 4, 5, 7 and 8) work satisfactorily by calling on the method (for example problem 4) from the main method.

Then you are done with the preparations😊

Task 4

Make a method problem4 in the Program class. In the method: read the contents of the file opg4.txt and print it (SOUT), line by line.

Task 5

Create a method task5 in the Program class. In the method: Receive 5 strings from the user (as you learned in step 4). Write the 5 strings of a file you call opg5.txt. Open the file and check if the result was as expected.

Tip: If you want to insert line breaks in the file, you can do so by entering the string "\ n".

Task 6

Create a "Artist" class that has fields: artistName, dateOfBirth (data type: LocalDate), city, country. Create a constructor for the class that accepts values for all the fields.

The constructor must set the values of the fields.

Make getters and setters for the fields. Also enter a method that prints (SOUT) information about the object (value for all fields).

Task 7

Create a method task7 in the Program class. The file opg7.txt contains data about artists in a specific format. Look at the contents of the file, and you will see what I mean. In the method (problem 7): Load the contents of the file and create Artist objects that match the data. Take care of the objects in a suitable structure.

To check if the reading from the file worked, go through all the objects and call the method that prints the object's condition.

Do you need any hints? Read the white text on a white background below:

Task 8

Create a method task8 in the Program class. In the method (problem 8): Read data from opg7.txt as you did in the previous problem. After the data has been read in, you change the values of some of the artists. Then write all the updated data (all the artists) to the opg8.txt file.

Open the file to verify that the changes were made.

Do you need any hints? White text on a white background below:

Extra task

Enter an additional field id for Artist. Make a copy of the file opg7.txt and name the copy extra.txt. Add a unique ID for each artist in extra.txt. Create a program that initially loads the artist data (extra.txt) from file. Remember that you must also read the ID. Create a user menu where the user can:

- Get information about all artists.

- Retrieve artist based on id.

- Add artist.

- Change artist based on id

- End the program.

When the user exits the program, the data about the artists must be written to file.

Extra difficult: When printing information about an artist, include information about how many years the artist is and how many days there are until the artist's birthday. We have not gone through this, so then you have to explore…